Extending Energetic Potential of Data Centers to Participate in Smart Grid Networks

Alexander Borgerding, Sven Rosinger

Extending Energetic Potential of Data Centers to Participate in Smart Grid Networks

Alexander Borgerding[1], Sven Rosinger[2]

[1]Carl von Ossietzky Universität Oldenburg, Ammerländer Heerstr. 114-118,
26129 Oldenburg, Germany
alexander.borgerding@uni-oldenburg.de
[2]OFFIS-Institute for Information Technology, Escherweg 2,
26121 Oldenburg, Germany
sven.rosinger@offis.de

Abstract. Data centers are growing due to the increasing demand of new and growing IT services. Following these trends, the electrical power consumption in data centers becomes a significant value. In parallel, an increasing share of renewable and volatile power sources needs to be handled in power networks due to the energy transition in Germany. To compensate the volatile behavior of renewables, appropriate actions are needed.

To take advantage of these issues, we present our approach to adapt the data center's power consumption. In our previous work, we pointed out the effects of applying different virtual machine allocation to data centers and to effect the server's power consumption. According to this approach, a controllable amount of power can be a valuable contribution to smart grid networks to keep power networks stable. In this paper, we propose our approach basing on server virtualization technology to adapt the data center's power consumption up to 50%. The approach is suitable in infrastructure as a service (IaaS) environments.

Keywords: Smart Grid, Data center, Server virtualization, VM placement, Energy efficiency, Power-aware, Resource management

1 Overview

On the one hand, the number of Information and Communication Technology (ICT) services increases year by year and existing technologies migrate into ICT services (telephone, TV, radio, etc.). Data centers are growing likewise, following these trends. On the other hand, the share of volatile renewable power sources is increasing due to the energy transition in Germany. This leads to volatile energy availability and in a consequence to varying energy price models. Furthermore, a challenging task emerges to ensure reliable and stable power network operations.

Data Centers are growing and their share of electrical power consumption becomes a meaningful value. This work investigates how data centers can participate in smart

grid networks. Their controllable amount of power consumption can be a valuable contribution to keep power networks stable.

Our Approach bases on server virtualization and applies different virtual machine allocations, inducing a specific server load and resulting in a dedicated power target. These allocations perform a forced overprovisioning of the server load, probably reducing Quality of Service (QoS) parameters like throughput, but keeping the services itself available. This allows a reduction of server's power consumption up to 50% for limited time periods.

This paper is organized as follows. In Section 2, we describe the current developments and the emerging need for action in connection with renewable power sources and growing data centers. Section 3 provides background information on methodologies in context of efficiency and adaptable power consumption in data centers. In Section 4, we introduce our approach to adapt the data center's power consumption. Section 5 presents the evaluation of the approach using the UPPAAL framework and obtained results. Finally, Section 6 concludes the main contents and discusses future extensions of our approach.

2 Problem definition

The renewable power sources are expanding and their part of the entire electrical power production is increasing. Most of the renewable power sources (i.e. solar and wind power) can not guarantee continuous power delivery. This emerges challenging tasks to keep power networks stable and to ensure reliable power network operations. To compensate volatile power behavior, further controllable power units are needed.

Following the demand of new and growing IT services, data centers are growing likewise. The part of electrical power, used in data centers, becomes a weighty part. In the same line energy costs raise steadily and data center operators are faced with questions about sustainability and efficiency. The idea now is to combine both developments to economic benefits.

In the context of IaaS environments, service details are defined in so called Service Level Agreements (SLA). In SLAs, customer and service providers define particular agreed aspects of the service i.e. quality, availability and responsibilities. Within these agreements, the service provider has a scope for economic action [4]. According to the agreements, the service provider can react to external power demands. In times of less energy availability suitable application can be slowed down.

The contractor provides server infrastructure only and does not have any direct access to the applications or knows further technical details. This is an impediment and hence, an applicable approach should be agnostic to the applications and effective without direct application access.

The target in this work is to thus develop an appropriate virtual machine (VM) allocation mechanism inducing the demanded power consumption within a minimum time period.

The process of accessing a VM allocation is comparable with the known knapsack or bin-packing problem. Objects (VMs) of different sizes (resource demand) must be

packed into a finite number of bins that correspond to the physical machines (PMs) of the same kind, while the number of used bins should be minimal. In case of VM allocation, several VM's resource demands have to be taken into account. Therefore, the task is related to a multidimensional bin-packing problem, while the bin-packing problem in itself has already NP-hard complexity.

In more precise, the number of possible VM allocations is described by partitioning a set of n elements into k partitions while the k sets are disjoint and nonempty. This depicts a datacenter with n VMs and k PMs and is described by the Stirling numbers of the second kind:

$$S_{n,k} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

For example, a server rack with 8 PMs and 40 VMs results in $S_{40,8} = 3{,}17 * 10^{31}$ different and possible VM allocations. Analyzing such amounts of eligible VM allocations takes disproportionate computation time. Hence, an all-embracing bin packing solver is not suitable to react on volatile power consumption demands. A heuristic online approach is needed to support short-term requirements. In this paper, we extend our idea of using competing VM allocations to take effect on server's power consumption.

The problem of allocating VMs related to the bin-packing problem can be formulated in a formal way. A data center with n VMs and k PMs is given. A set of VMs $\mathrm{V} = \{\mathrm{VM}_1, \dots, \mathrm{VM}_n\}$ represents the objects and a set of PMs $\mathrm{P} = \{\mathrm{PM}_1, \dots, \mathrm{PM}_k\}$ represents the bins. The VMs are represented as resource demand vectors d_i and the PMs are represented as capacity vectors c_s.

For an efficient PM utilization, the task now is to find a VM allocation that maximizes j, while j is the number of VMs operated on the PM and for all PMs in the set of P the following applies:

$$\sum_{i=1}^{j} d_i \leq c_s$$

To assess the quality of a VM allocation, the efficiency of a VM allocation is a suitable metric [9]. The efficiency E of a VM allocation A is described as:

$$E(A) = \frac{work\ done}{unit\ energy}$$

3 Related work

The subject of controllable power consumption in data centers is a topic on which a lot of work has already been done. In this part of the paper, we describe different approaches in the context of controllable power consumption, energy efficiency and integrating volatile power availability in data centers.

Most of the existing approaches rely on application task scheduling mechanisms [1, 20] and base on the fact that the power consumption of a server corresponds to its utilization. To approach the needed power consumptions, [1] uses workload schedulers. A task list is generated, where the tasks are sorted by their execution deadline. The task with the earliest deadline is at the top of the list. If energy is available, the scheduler starts task from the list, beginning at the top. This is an earliest deadline first (EDF) scheduler. In times of power shortage, the scheduler simply terminates task to reduce the power consumption. This methodology requires detailed information about the applications to build the EDF task list. If tasks needed to be terminated or deadlines exceeded, appropriate measures are required.

To increase efficiency of data centers, Tang et al. [20] improved the data center's cooling efficiency. A central EDF Scheduler is set up, placing tasks on the coldest server in the data center. This avoids the occurrence of hot spots and cooling requirements can be decreased. But a task scheduler comes with the same disadvantages, as described. In summary, those approaches are not applicable in the context of IaaS environments due to the absence of application access.

Other approaches make use of VMs as migratable workload container [7, 13, 15]. Corradi et al. [11] make use of VMs as container to migrate workload from one server to another. The VM is a suitable instrument to avoid dealing with application-internal topics. They concentrate VMs on a fewer number of servers to power down released servers in order to improve the overall efficiency. Casale et al [7], Kumar et al [13] and Pu et al. [22] use virtualization technology to improve efficiency with attention on server resource correlation. Evoking specific power consumption is not a target of their work.

One concept to adapt a data center's power consumption is to migrate server workload from one data center to an other. There are approaches [8, 14, 16] that make use of varying local energy availability and migrate server workload across different data center destinations. The power consumption is transferred followed by the workload migration but this methodology is only suitable for wide-spread scenarios and the actual power consumption is not effected at all. Data center locations at close quarter typically have similar energy conditions.

In summary, existing approaches in connection with volatile energy availability and data center power consumption are based on variations of scheduling solutions. Application tasks are operated in times of energy availability and are interrupted otherwise. All these approaches are application-invasive and technical details are needed to schedule the tasks. Hence, they have to deal with application specific issues as impediment. An application-independent, agnostic approach, offering controllable server power consumption is still missing.

In our previous work [6], we pointed out the usability of developing different VM allocations to affect application's performance and server's power consumption as well.

4 Approach

Modern data centers use server virtualization technologies to optimize operating services [4]. Virtualization technologies allow live migration of virtual machines (VM) and service levels can be improved by moving applications seamlessly from one server to another [4]. To avoid application-internal issues, our approach focuses on VM environments.

In our previous work, we pointed out that the data center's power consumption is dependent on different VM allocations [6]. Various VM allocations effect different PM utilizations leading to corresponding server power consumptions. In this paper, we present our approach to induce a specific target server power consumption, basing on the knowledge gained. Our approach intends to extend the possibilities of power control in data centers. The goal is to provide recommended electrical power consumption within an expected amount of time and, in addition, to keep the efficiency at a high level.

To the best of our knowledge, this is the first efficiency-aware approach that researches on agnostic methodologies, to control the data centers power consumption.

4.1 System model

Our methodology applies different VM allocations in order to reach different, desired server power consumptions. Fig. 1 shows a concise system overview containing PMs equipped with an optimizer component and operated VMs.

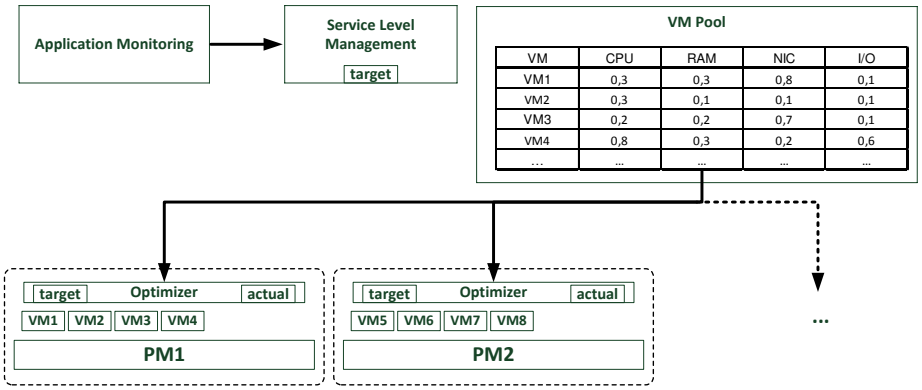

Fig. 1. Schematic System Model

The Application-Monitoring component collects technical information about the application's health and PM's status.

The Service Level Management (SLM) component collects the delivered information and calculates – depending on the Service Level Agreements (SLA) – new target power consumption values. Determining target values is a task without any local PM dependencies and can be done globally by the SLM component. We do not

focus the task of assessing the SLAs; our point is to provide the required target power consumption.

The optimizer component receives periodically updated target information from the SLM and compares it with the actual power consumption. Afterwards, the optimizer initializes VM permutations, if necessary, to reach the new target by choosing and migrating VMs from the entire VM Pool that fit best. The VM Pool provides information about the VM's resource demand (Central Processing Unit (CPU), Random Access Memory (RAM), Network Interface Card (NIC), and Input/Output operations (I/O)). This information is essential to estimate a VM migration and to obtain the utilization of a PM by summarizing the resource demands of all VMs allocated to a PM.

The resulting VM allocation induces a PM utilization evoking the required PM power consumption. The methodology of choosing VMs bases on the creation of VM subsets and furthermore analyzing their resource requirements as it is described in the following Section 4.2. A specific VM allocation can be used to explicitly reduce overall PM's resource utilization by combining resource-competing VMs, leading to lower server power consumption and vice versa.

4.2 VM allocation algorithm

As described, the problem of determining a VM allocation has NP-hard complexity. To reduce the complexity and long computation times, we do not focus on an all-embracing, global solution. The task is split into subtasks to be solved by the optimizer component, working on each of the PMs. The optimizer focuses on a solution for just its own PM with little or no loss of efficiency.

For further reducing the complexity, we divide the servers into two groups, as shown in Fig. 2.

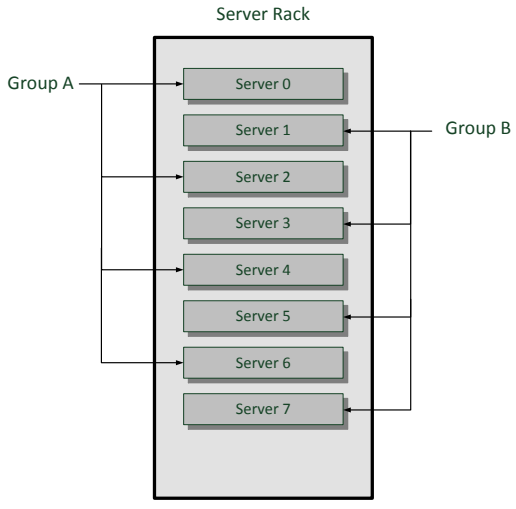

Fig. 2. Servers divided into two groups

We get a constellation with alternating order of servers belonging to group *A*, with even server IDs and group *B*, with the odd server IDs. Increasing the server's power consumption is assigned to server group *A* by migrating VMs from PMs of group *B* to PMs of group *A*.

Therefore, a subset of migration candidates from group *B* to group *A* is built. Suitable VMs to increase the power consumption have to fulfill several conditions. First, the VM's RAM resource demand has to fit to the target PM of group *A*. Secondly, the actual VM's host PM within group *B* should have a CPU utilization of more than 100% and, thirdly, the resulting CPU utilization of the target PM in group *A* including the potential VM candidate should not exceed 100%. Such a migration of a VM from an overloaded PM of group *B* to an underutilized PM of group *A* will result in an overall power consumption increase. Generating CPU utilization above 100% of the target PM reduces the potential for further power consumption increases.

The subset of possible and randomized selected candidates will contain up to five VMs, fulfilling the described conditions. Based on this set, the VM is chosen for migration that completes the CPU utilization of the target PM near to 100%.

Server group *B* focuses on decreasing the power consumption. It starts with building a subset of all VMs, analog to the procedure of server group *A*. The conditions for suitable VMs are fitting RAM resource demands and that the VM is actually operated on a PM that belongs to server group *A*. In the last step, the VM with the greatest CPU utilization demand is chosen and migrated onto the PM of group *B*. This VM offers the best reachable effect per VM migration.

As additional effect of dividing the system into two groups, we got equilibrium server utilization within the groups; this prevents the occurrence of hot spots and leads to increased cooling efficiency similar to the approach of Tang et al. [20].

Our approach bases on developing resource competing VM allocations and resolving resource competing VM allocations to decrease and increase the server's power consumption. In our previous work, we pointed out the effects of applying these VM allocations [6]. With focus on efficiency, we analyzed server components and their behavior regarding different resource utilizations. A server resource is sought by the directive to develop efficiency-aware resource competing and resource over-provisioning VM allocations.

RAM over-provisioning is not an option because virtual RAM pages are stored on the hard disk and the additional process of loading and storing RAM pages on the hard disk in itself leads to efficiency loss. Similar effects arise by over-provisioning the hard disk. Parallel writing and reading needs additional repositioning of the disk-heads and leads obviously to reduced efficiency and may evoke bursty workloads [7].

We tested CPU efficiency in case of over-provisioning. In experiments we evaluated the CPU as being a potential server resource for over-provisioning. Our experiments show a linear behavior when benchmarking the CPU.

Fig. 3 shows the results of our experiments. In six test cases we applied a CPU benchmark that handles 10.000 events. In the first test case we operate one VM executing the benchmark on the PM. It needs about 30 sec.

In the second test case we operate two VMs in parallel on the PM running the benchmark and it needs about 60 sec. Three VMs in parallel needs about 90 sec. and

so on, as Fig. 3 shows. We observe a linear behavior of the CPU, as the output and duration times show. Furthermore, a single execution of the benchmark produces the same events per second as several executions in parallel.

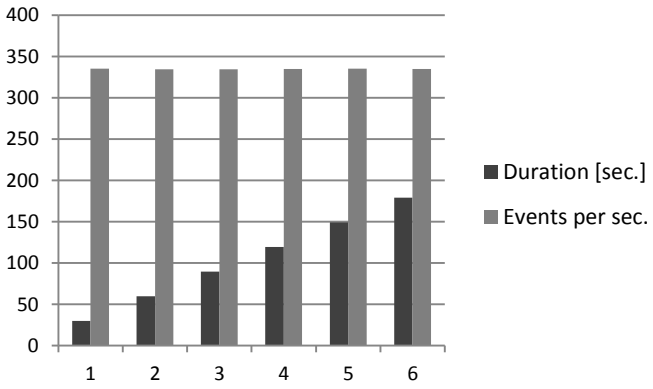

Fig. 3. CPU efficiency test

To conclude, over-provisioning the CPU is not resulting into any noticeable efficiency loss. And in addition, we could not discover a significant effect if we spread the workload to several VMs. As expected, the server's CPU resource is suitable to develop competing VM allocations and to slow down applications without efficiency loss.

5 Evaluation

To evaluate the approach, we implemented a simulation based on timed automatons. We use the UPPAAL platform as mentioned in [12] for modelling the system behavior as automaton-system with all VMs and PMs represented as individual and parallel simulated automatons. The resulted environment allows us to explore different configurations and system behavior in simulations. Existing frameworks i.e. [3] do not provide continuous VM operations and effects of intended lowering applications are not considered.

5.1 VM automaton

Fig. 4 shows the UPPAAL automaton, representing the behavior of a VM. The edge from the initial location initializes the VM. The edge from location `initialized` to `operate` includes synchronization while the VM is waiting for a PM that is going to operate it. The PM sends a signal via synchronization `ini[id]`, while `id` is the internal unique id of the VM. The VM changes to operation displayed by the location `operate`. Now, the VM is initialized and in normal operation mode.

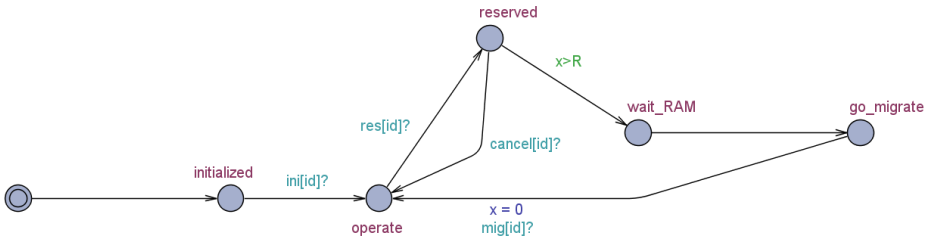

Fig. 4. VM automaton

In case of a planed migration, a PM reserves the VM via `res[id]`. The VM is reserved now; no other PM can reserve this PM. But, a reservation can be canceled by sending a signal via channel `cancel[id]`.

Before starting the preparation for migration, the next edge is equipped with a guard. To prevent frequent migrations and to guarantee a minimum level of operation time the VM will not enter the migration process until the time `R` is passed. The value of `R` is determined according to SLA definitions. The clock `x` is started at the last migration of the VM and will be compared with the value of `R`. If the operating interval is longer than the value of `R`, the next location is reachable.

The migration of a VM is divided into several steps. In the first step, the RAM content needs to be copied to the new target PM. According to the copy process, the `wait_RAM` location will be left a far as the time to copy the RAM is passed. In the last step of the migration, the PM signals the completion via channel `mig[id]` and the clock `x` is reset.

5.2 PM automaton

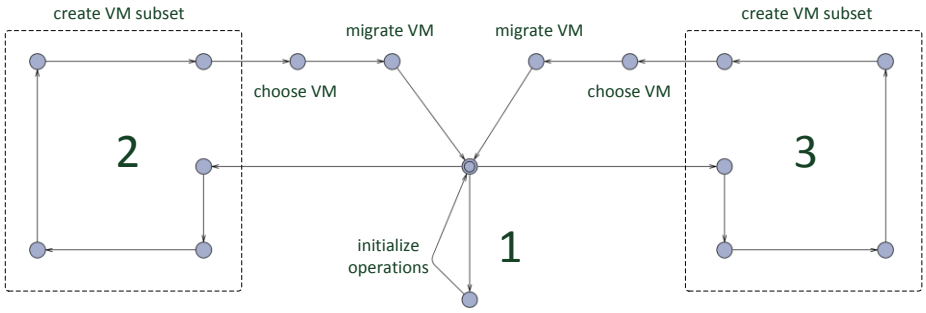

Fig. 5. PM automaton

Fig. 5 shows a schematic illustration of the automaton representing the PM. The entire system consists of two groups of PMs, every PM belongs to one of these groups. Server group *A* acts in case of increasing and group *B* in case of the need of decreasing the power consumption as described in Section 4.2 of this paper.

In general the automaton of a PM consists of three branches. Branch 1 ensures the initial VM operation. It is entered if VMs exist that are not in operation mode at all.

As long as the system contains initialized and not operated VMs, the PM takes VMs to operate until the PM reaches its resource limit.

The branches 2 and 3 are entered, if the need to increase or decrease the power consumption emerges. Branch 2 is responsible for increasing power consumption and represents the behavior of server group A. The branch 2 is entered, if the power consumption is lower than the target value and the PM's CPU utilization is lower than 100%. If the CPU utilization is already higher, hosting additional VM will not increase the power consumption. In the next steps, up to five VMs are chosen, fulfilling the following requirements:

- VM's RAM resource demand fits to the PM
- VM is actual operated on a PM belonging to group B
- Actual VM's PM has got a CPU utilization more than 100%
- VM candidate's CPU demand + actual PM's CPU utilization is below 100%

The VM with the best fit to 100% CPU utilization is chosen; and finally, the VM is migrated onto the PM.

Branch 3 is entered for decreasing the power consumption and representing the behavior of the odd server group B. The branch is entered, if the actual power consumption exceeds the power consumption target value. Analogous to the methodology of group A, a subset of five VMs is built. Potential candidates are actual operated on PMs belonging to group A for the reasons already mentioned. Furthermore, the resource demand, except CPU demand, fits to the PM. The VM with the largest CPU demand is chosen and migrated onto the PM as described in Section 4.2.

The UPPAAL framework provides the possibility to implement program code similar to the program language C. We make use of this feature to implement several functions to calculate resource capacities i.e. and to choose the VM candidates, as described. Moreover, we sum up the resource demands of all VMs allocated to a PM to calculate its utilization. The utilization again is the key value to calculate the PM's power consumption [17]. We map the resulted utilization to a lookup table containing utilizations and corresponding power consumptions of the simulated PM at a granularity of 10%-steps.

5.3 Results

The results shown in Fig. 6 were obtained in a simulation scenario with 40 VMs and 8 PMs. The simulated PMs are servers of type Fujitsu Server PRIMERGY TX2560 M1.

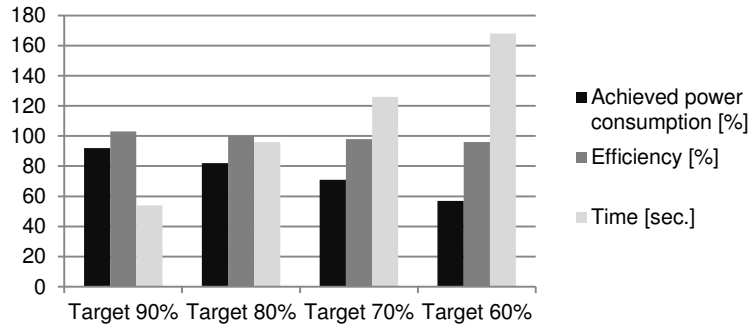

Fig. 6. Achieved power consumption results

The power consumption values are calculated basing on data sheets [19] from the Standard Performance Evaluation Corporation (SPEC). Fig. 6 shows results of demanding power consumption targets of 90%, 80%, 70% and 60% of maximum power consumption (100%).

The simulation run with a target of 60%, for example, shows a reduced server power consumption from 2112W to 1260W according to an achieved power consumption of 57%. We achieved power consumptions with only a slight discrepancy from the target, while at the same time the efficiency is kept at high level. The time bar illustrates the summed migration times. Obviously, more migrations are needed to achieve a greater amount of power reduction.

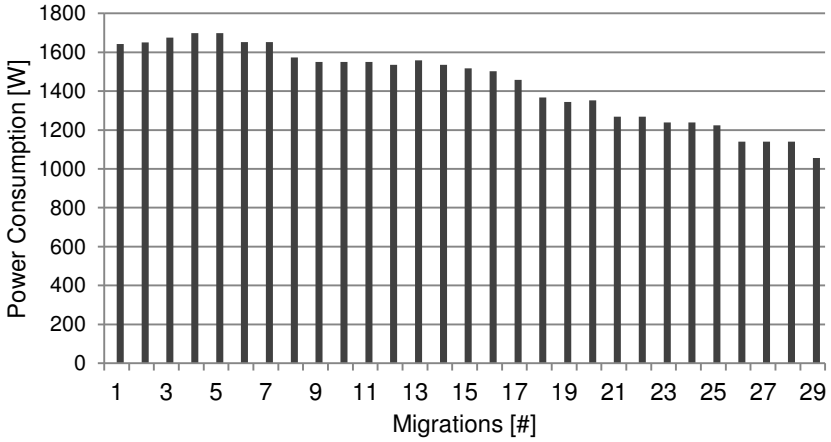

Fig. 7. Power consumption during migration

In Fig. 7 the results of a simulation with target of 1000 Watt power consumption is shown. 29 VM migrations were needed to reach the target. The power consumption is not decreasing monotonously in every migration step. This is caused by the characteristic power consumption of the servers. The server's output performance is not linear regarding the different server utilizations.

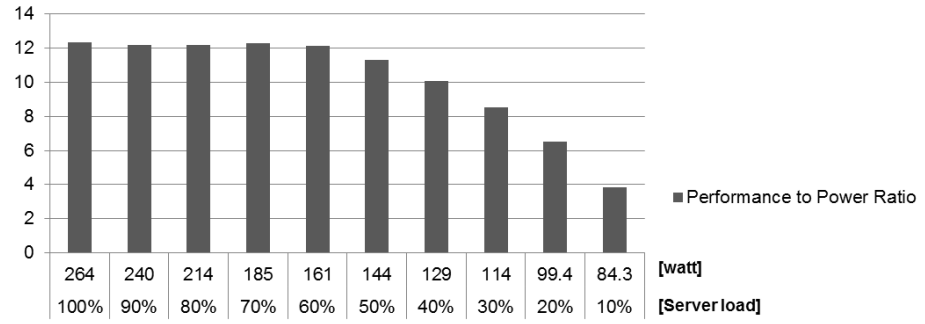

Fig. 8. Power to Performance Ratio [19]

Fig. 8 displays the Performance to Power Ratio belonging to the simulated PM Fujitsu Server PRIMERGY TX2560 M1. Performance to Power Ratio means the average throughput divided by the power consumption.

Below a utilization of 60% the ratio is decreasing and the efficiency is shrinking. Our Approach is migrating VMs from one PM to another. While the migration processes, we get VM allocations that cause PM utilizations with lower efficiency, leading to temporally increasing power consumptions. This effect relates to the different efficiency levels of the PM.

When increasing the server's power consumption, servers belonging to group *A* are released from hosting VMs. Hence, the servers of group *A* are passing through lowered efficiency levels, leading to temporally higher power consumption.

Larger scenarios with an increased number of servers can be divided in several sub installations of our approach. Hence, every data center can be equipped with our methodology and we got transferable results.

The actual data center's power consumption in Germany is about 10 TWh per annum [5]. According to 10 TWh per annum, the average total data center power consumption is about 1141.5 MW. Regarding this amount of power consumption, and if our approach is applied to only a half of these servers, it would result into controllable power consumption of 228.3 MW, which is available within a timeframe of about 180 seconds.

The grid control cooperation (GCC) is an innovative network control concept, by means of which the four German transmission system operators (TSOs) optimize their control energy use [18]. In this context, the so-called secondary control reserve (SCR) is an amount of power that has to be available within 5 minutes. In comparison, the desired negative SCR of the TSO EnBW Transportnetze AG is about 311 MW [10]; our approach can provide 228.3 MW of negative SCR.

6 Conclusion and future work

We described the emerging needs to adapt the data center's power consumption and figured out that a non-invasive approach regarding the applications that is suitable for IaaS environments is still missing.

Our approach delivers a methodology to control the data center power consumption and enables data centers to be a valuable member of smart grid networks. It operates without direct application access and offers decreasing power consumption potential of about 50%. This is similar to approaches depending on scheduling technologies [9].

The RAM-size of a PM is a limiting factor while condensing VMs onto the PM. Barker et al. [2] pointed out the potential of memory sharing. Wood et al. [21] developed a fingerprinting system to determine the sharing potential among a set of VMs. Extending the migration policies to consider RAM sharing potential might improve our approach and will be part of our future work.

References:

1. Aksanli, B., et al.: *Utilizing green energy prediction to schedule mixed batch and service jobs in data centers.* SIGOPS Oper. Syst. Rev. 45, 3 (January 2012), pages 53-57, 2012.
2. Barker, S., et al.: *An empirical study of memory sharing in virtual machines*. In Proceedings of the 2012 USENIX conference on Annual Technical Conference (USENIX ATC'12). USENIX Association, Berkeley, CA, USA, 25-25, 2012.
3. Beloglazov, A. and Buyya, R.: *OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds*, Concurrency and Computation: Practice and Experience, 2014.
4. Bobroff, N., et al.: *Dynamic Placement of Virtual Machines for Managing SLA Violations*, 2007 10th IFIP/IEEE International Symposium on Integrated Network Management, Munich, pages 119-128, 2007.
5. Borderstep Institut für Innovation und Nachhaltigkeit gemeinnützige GmbH: *Deutliches Wachstum bei deutschen Rechenzentren – Update 2015*, https://www.borderstep.de/wp-content/uploads/2015/01/Borderstep_Rechenzentren_2015_Stand_16_12_2015.pdf, 2015. Accessed 10 Jan 2017.
6. Borgerding, A. and Schomaker, G.: *Extending Energetic Potentials of Data Centers by Resource Optimization to Improve Carbon Footprint*. Advances and New Trends in Environmental and Energy Informatics. Springer International Publishing, pages 3-19, 2016.
7. Casale G., et al.: *A Model of Storage I/O Performance Interference in Virtualized Systems*, 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, pages 34-39, 2011
8. Chen, C., et al.: *Green-aware workload scheduling in geographically distributed data centers*. 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, pages 82–89, 2012.
9. Chen, H., et al.*: The data center as a grid load stabilizer*, 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore, pages 105-112, 2014.
10. CONSENTEC Consulting für Energiewirtschaft und -technik GmbH: *Gutachten zur Dimensionierung des Regelleistungsbedarfs unter dem NRV*, http://www.consentec.de/wp-content/uploads/2012/01/Gutachten_zur_Hoehe_des_Regelenergiebedarfes_2010.pdf, 2010. Accessed 10 Jan 2017.
11. Corradi, A., et al..: *VM consolidation: A real case based on OpenStack Cloud*, Future Generation Comp. Syst., 32, pages 118-127, 2014.
12. David, A., et al.: Model-Based Framework for Schedulability Analysis Using UPPAAL 4.1. Model-Based Design for Embedded Systems, pages 93-119, 2010

13. Kumar, P., et al.: *Power and Data Aware Best Fit Algorithm for Energy Saving in Cloud Computing*, International Journal of Computer Science & Information Technology; Vol. 5 Issue 5, p6712, 2014.
14. Krioukov, A., et al.: *Integrating renewable energy using data analytics systems: Challenges and opportunities.* IEEE Data Eng. Bull., 34(1), pages 3–11, 2011.
15. Liu, H., et al.: *Performance and energy modeling for live migration of virtual machines*. In Proceedings of the 20th international symposium on High performance distributed computing (HPDC '11). ACM, New York, NY, USA, pages 171-182, 2011.
16. Liu, Z., et al.: *Greening geographical load balancing*, ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '11, New York, USA, pages 233–244, 2011.
17. Pelley, S., et al.*: Understanding and abstracting total data center power*. Proc. of the 2009 Workshop on Energy Efficient Design (WEED), Jun. 2009.
18. REGELLEISTUNG.NET Internetplattform zur Vergabe von Regelleistung*: Grid Control Cooperation (GCC)*, https://www.regelleistung.net/ext/static/gcc?lang=en. Accessed 10 Jan 2017.
19. Standard Performance Evaluation Corporation (SPEC), Benchmark Results Summary of Fujitsu FUJITSU Server PRIMERGY TX2560 M1, https://www.spec.org/power_ssj2008/results/res2015q2/power_ssj2008-20150317-00689.html, 2015. Accessed 10 Jan 2017.
20. Tang, Q., et al.: *Thermal-Aware Task Scheduling for Data centers through Minimizing Heat Recirculation*, The IMPACT Laboratory School of Computing and Informatics Arizona State University Tempe, AZ 85287, 2008.
21. Wood, T., et al.: *Memory buddies: exploiting page sharing for smart colocation in virtualized data centers*. In Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments (VEE '09). ACM, New York, NY, USA, pages 31-40, 2009.
22. Pu, X., et al.: *Who Is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds*, IEEE Transactions on Services Computing vol. 6 no. undefined, pages 314-329, 2013.